Fault identification through the combination of symbolic conflict recognition and Markov Chain-aided belief revision

Shen, Qiang; Smith, Finlay

Published in:
IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)

DOI:
10.1109/TSMCA.2004.832826

Publication date:
2004

Citation for published version (APA):
Shen, Q., & Smith, F. (2004). Fault identification through the combination of symbolic conflict recognition and Markov Chain-aided belief revision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *34*(5), 649-663. https://doi.org/10.1109/TSMCA.2004.832826

Fault Identification Through the Combination of Symbolic Conflict Recognition and Markov Chain-Aided Belief Revision

Finlay S. Smith and Qiang Shen

***Abstract*—Fault identification is a search for possible behaviors that would explain the observed behavior of a physical system. During this search, different possible models are considered and information about the interaction between possible behaviors is derived. Much of this potentially useful information is generally ignored in conventional pure symbolic approaches to fault diagnosis, however. A novel approach is presented in this paper that exploits uncertain information on the behavioral description of system components to identify possible fault behaviors in physical systems. The work utilizes the standard conflict recognition technique developed in the framework of the general diagnostic engine (GDE) to support diagnostic inference through the production of both rewarding and penalizing evidence. In particular, Markov matrices are derived from the given evidence, thereby enabling the use of Markov chains to implement the diagnostic process. This work has resulted in a technique, which maximizes the use of derived information, for identifying candidates for multiple faults that is demonstrated to be very effective.**

***Index Terms*—Belief updating, conflict recognition, Dempster–Shafer, fault identification, general diagnostic engine (GDE), Markov chains.**

I. Introduction

THE ABILITY to successfully and efficiently diagnose faults in physical systems is important in ensuring that the physical systems can be repaired and returned to use as quickly as possible. The automatic detection and diagnosis of faults cannot only speed up this process, but also free the human engineers to concentrate on maintaining the physical systems (for example, the tiger system for gas turbines [1]). A key aspect of the diagnostic process is fault identification, which involves searching for possible faults that could explain the observed faulty behavior of the physical system. Fault identification is essentially a search through the search space of all possible faults, looking for combinations of faults that could explain the observed behavior [2]. Fault identification techniques involve a guided search through this search space, with the success of the technique being measured in terms of the efficiency of the search process.

Manuscript received July 8, 2002; revised May 6, 2003 and February 12, 2004. This work was supported by the UK EPSRC under Grant 96307135. This paper was recommended by Associate Editor H. Pham.

F. S. Smith is with the Department of Information Technology, National University of Ireland, Galway, Ireland (e-mail: finlay.smith@nuigalway.ie).

Q. Shen is with the Centre for Intelligent Systems and Their Applications, School of Informatics, University of Edinburgh, Edinburgh, EH8 9LE, UK (e-mail: qiangs@inf.ed.ac.uk).

Digital Object Identifier 10.1109/TSMCA.2004.832826

A diagnostic process can be viewed as deriving a model, which is consistent with all of the observed values, forming the fault hypothesis. For many domain systems, there exists some knowledge, both of how individual components may fail and of how likely these failures are even though such knowledge may well be incomplete, imprecise and/or uncertain. Many diagnostic systems use this information to aid the diagnostic processes. For example, information about system behavior can be used to compile off-line guidance using existing knowledge [3]. However, the use of knowledge about how likely individual components are liable to fail may be difficult [4] as not only can the definition of failure probability make the values very subjective, but also such data may not be completely available.

The work presented in this paper is based upon existing preliminary research [5], [6], which only dealt with diagnosing faults in very simple systems. It exploits the simplicity of a general diagnostic engine (GDE)-style [7] candidate proposer in utilizing an assumption-based truth maintenance system (ATMS) [8] to recognize conflicts and makes use of the basic principles of Markov chains [9] to identify faults by manipulating beliefs in normal, faulty, or unknown behavior models. Here, a normal behavior for a component is one where the component behaves as it was designed, a faulty behavior is one where the component does not behave as designed, and the unknown behavior is a fault that is not known to the diagnostic system.

It is well recognized that as the values of the failure probabilities are unlikely to be either accurate or complete, diagnostic algorithms that use them must not be too dependent upon the exact probabilistic values. The work described herein overcomes the difficulties involved in obtaining exact data as it is not dependent on such exact values. Having accurate and complete failure probabilities would improve the overall efficiency of the work described in this paper, however, if such comprehensive and accurate data were not available the effect on the performance would not be significant. For example, if for a given component, the failure probabilities were not known, then all of the faults would be assigned the same failure probability. If that component then failed, the possible faults would be considered in a random order rather than considering the most likely fault first. If some uncertain knowledge of the failure probabilities were known, such as the most likely way for a component to fail, this would improve the performance of the algorithm as the most likely faults would be considered first. The algorithm is therefore robust enough to cope with incomplete or inaccurate failure probabilities.

As with many other approaches to model-based diagnosis [10], a complete system model in the present work consists of a set of behavioral descriptions of the system components and the structural description of the interactions between the components. Each component is modeled as having one of a fixed set of possible behaviors, which are independent from each other. The set consists of one normal behavior and the others faulty, including the unknown behavior that captures all previously unexperienced faults. The belief revision method described in this paper assumes independence. The requirement for these behaviors to be independent of each other is perfectly reasonable as evidence for or against a particular behavior does not directly refute or confirm any other behavior(s), since the evidence (relating to a given component) is based upon the effect of a single behavior and in no way discriminates between any of the other possible behaviors (for that component).

Behavioral descriptions are herein referred to as model fragments [11] of the component concerned. These model fragments will have a belief attached, reflecting how likely a particular fragment is of being the fragment that describes the actual behavior of the corresponding component. The use of Markov chains in the revision of such beliefs provides an efficient mechanism that is based on rigid mathematical principles. The result is a diagnostic process with the capability of belief management, which can use prior knowledge where it exists without a loss of generality and which, when linked to the detected discrepancies, can offer important information for postulating likely fault behaviors.

Section II introduces the theoretical background of the present work, including Markov chains and the symbolic conflict-recognition method used in GDE. This is followed by a section that gives a method for using a form of Markov chains to update beliefs, and another that describes a method for using these updated beliefs in a GDE type diagnostic system. Section VI reports on experimental results of fault identification, while Section VII presents conclusions.

II. Background

This section outlines both Markov chains and GDE's conflict-recognition procedure.

A. Fault Diagnosis

The diagnosis of faults in physical systems has been the subject of much work over recent years [12]–[14], both in static systems and dynamic systems. This section focuses on the diagnosis of faults in static GDE-type systems.

1) GDE-Type Diagnostic Systems: GDE is used to diagnose multiple faults in static electrical circuits (though applications of GDE to other domains have also been reported e.g., [15] and [16]). Fig. 1 shows an example of a simple circuit, which is often used to illustrate the ideas of GDE.

The GDE makes several assumptions: 1) multiple faults can occur; 2) the model represents the structure and behavior of the physical system; and 3) a component either works or it does not. GDE also assumes that faults must occur in components, which is not as limiting as it first appears as connections (e.g., wires in electrical circuits) can also be represented as simple components.

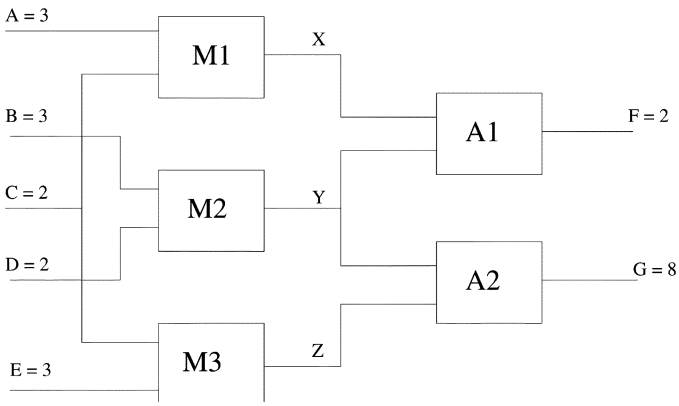

Fig. 1. Illustrative diagnostic problem.

The method employed by the GDE is to use given variable values, typically the exogenous variables to propagate through the model, allowing predicted values to be generated for each of the indigenous variables. Whenever two *different* values are predicted for the same variable, a conflict exists as a variable cannot have two different values simultaneously. Subsequently, as it is known how the two values were derived, at least one of the components that was used to predict either of the two values must be faulty. The combination of the two sets of components that led to the derivation of conflicting values is called a conflict set.

The major limiting factor with the original GDE is that it only considered models that were either working normally or not working normally. This can lead to obviously inappropriate candidate sets. This limitation of the GDE was part of the motivation behind several of the improvements that were made to the GDE [10], [12], [17], [18]. The other main motivation behind these improvements to the GDE was that, in general, some possible known fault behaviors exist. The aim of these extensions was to combine these known fault behaviors with the relative simplicity and power of GDE to create more effective diagnostic processes.

Sherlock: A particular and successful extension to GDE is Sherlock [12], [18]. Rather than trying to rule out fault behaviors, Sherlock attempts to rule out all known behaviors, including the normal behavior. This leads to the possible situation where all of the possible behaviors of a component have been eliminated. Sherlock handles this by adding a new type of behavior, the unknown behavior. If all of the other possible behaviors have been eliminated, Sherlock deduces that the component must be behaving in some previously unknown manner. This additional behavior allows Sherlock to avoid having to know all of the possible fault behaviors of a given component, rather Sherlock only needs to know the most likely fault behaviors. This greatly reduces the search space, as only a relatively small number of possible behaviors need to be specified.

Another feature that Sherlock uses is that of failure probabilities. Each possible behavior (including the normal behavior and the unknown one) is assigned an initial probability. These probabilities are used to select fault candidates. The probabilities are

revised based upon the observed values, this has the effect of focusing the search process to the components in the model that are most likely to be faulty.

When Sherlock generates conflict sets, it uses them to guide the search process by adjusting the probabilities and then disposing of them. This information on conflicting observations and predictions is never used again, though it may contain useful information.

B. Basic Principles of Markov Chains

The fundamental principle of Markov chains is that the current state of a variable is dependent only upon the immediately previous state, with no dependence upon other previous states [9]. In fault diagnosis, variables represent which behavior each component is currently displaying and the states of the possible behaviors (including the unknown one) of each of the components. The transition between states in the diagnostic process therefore represents the selection of the next fault candidate. In this context, it is not important how the diagnostic process reached a particular state, only how it is going to select the next fault candidate.

Markov chains provide a powerful tool for evaluating the probabilities of future events based only on the current state, with the resultant probabilities indicating how likely each of the potential successor states are. However, the way in which the transition probabilities and the initial state-probability vector are obtained is, in general, very domain dependent, with the values having to be accurate to ensure that any subsequent analysis is accurate. Fortunately, in the present application, only the beliefs, not the probabilities, of which model fragment represents the actual behavior of a component is considered, and the evidence used is very approximate, while the accuracy is not so important. This is acceptable as the relative values of the beliefs relating to each component is only intended to indicate a ranking of each possible behavior rather than an absolute probability.

Related Use of Markov Techniques in Monitoring and Diagnosis: There is some limited existing work in utilizing Markov chains in fault diagnosis. In [19], a modeling language that was developed using Markov processes and qualitative modeling techniques has been consolidated into a diagnostic process that includes belief revision. The use of Markov processes in the modeling language allows for a simulation that not only predicts the future states of a physical system, but also measures how likely each of these future states are. The diagnostic process uses the observations of the physical system together with control actions to revise the beliefs in individual states to help the process of selecting fault candidates. This work contrasts with the work presented in this paper, in that the Markov chains are used in the modeling and simulation of the system, whilst the work here uses Markov chains to revise beliefs in individual components and so aids in the process of selecting fault candidates.

Another related approach is to use Markov processes to predict the values of unmonitored dynamic variables [20]. The outcome is an aid for model simulation that is tolerant to noise and shows promising results, even with fairly approximate system models. This, again, contrasts with the belief revision process described in this paper.

Earlier related work in model simulation used Markov chains in a qualitative simulator [21], that used a large, sparse, Markovian matrix to predict the next state given the current one. The difficulties with this approach are that the matrix itself could be enormous (even allowing for it being sparse) and all possible state transitions need to be known in advance.

III. Belief Updating

The work presented in this paper utilizes beliefs in individual model fragments to select fault candidates. Each component within the model of the physical system has several possible behaviors. Each model fragment (for each component) has a belief associated with it, the higher the belief the more likely that the model fragment represents the observed behavior of the component. As the diagnostic process generates evidence for and against individual model fragments, a method is required of updating the beliefs in individual model fragments. This revision is necessary to support that only the most likely model (based upon all of the evidence) is selected. Note that the initial beliefs in the model fragments may either be derived from expert experience or be set in a similar fashion to the way that Sherlock [18] assigns initial probabilities.

During a diagnostic session, every time a model is simulated, conflict and confirm sets are generated. A conflict set is generated when two different values of the same indigenous variable can be simulated. A confirm set is generated when two identical values of the same indigenous variable can be simulated in different ways. These conflict and confirm sets are then used to generate penalizing evidence and rewarding evidence, respectively, for and against individual model fragments. Given the new evidence the values of belief in individual model fragments should be revised. Markov chains are ideal for handling this problem, as the revision is only based upon the evidence and the current beliefs. What is therefore required is a means of converting the given evidence into a Markov matrix.

1) Updating Beliefs: To simplify the description of the belief-updating process, the explanation that follows only deals with the beliefs in a set of model fragments relating to a single component. The process has to be repeated for every component in the system for a complete revision of the beliefs, as each component has its own sets of beliefs and evidence. An important advantage of updating the beliefs in the fragments of each component separately is that the complexity of the revision process only increases linearly with respect to the number of components, as each component requires the same amount of computation so doubling the number of components doubles the complexity.

As indicated earlier, there are two types of evidence: rewarding or penalizing. Negative evidence is effectively treated as positive evidence for the hypothesis that the component is not behaving as expected (in other words that one of the other behaviors is the actual behavior). Let E_i be evidence that relates to fragment i of a given component, $E_i \in [-1, 1]$, where a positive value of E_i signifies a piece of rewarding evidence and a negative value represents a piece of penalizing evidence. In particular, a value of 1 means that the fragment definitely explains the current behavior of the physical component and

a value of -1 means that the fragment definitely does not explain the current behavior. The size of a piece of evidence is a measure of the strength of that evidence, that is the closer the evidence is to 1 (or -1) the stronger the belief in the evidence.

A piece of penalizing evidence is calculated in the diagnostic process based upon the conflict sets that are generated during the conflict recognition phase. Small conflict sets lead to strong negative evidence, as usually the smaller the conflict set the more likely that a given member of the set is faulty, therefore fewer members in a conflict set reflect stronger evidence against each of the members of the conflict set. If a fragment appears in more than one conflict set the evidence against it increases. Confirming evidence is calculated in a similar way, based upon the confirm sets that have been generated.

Intuitively, the revision process must reflect the strength of the evidence so that a value closer to 1 (or -1) results in a relatively larger change in the beliefs, whilst a value closer to 0 results in a relatively smaller change. In addition, if the evidence is rewarding, the belief in that fragment will have to be increased and if the evidence is penalizing, the belief in that fragment will have to be reduced. In converting given evidence to a Markov matrix, these two kinds of evidence are dealt with separately as the effects of the two kinds of evidence are so different.

Explicitly representing each possible model at system level as a possible state and calculating the probabilities of each possible transition between such states would be an immense task for systems of a realistic complexity. For a simple system of say, four components, each of which has five possible behaviors, would require $5^4 = 625$ explicit representations. Even for such a simple system, the Markov matrix is very large: 625×625. To overcome this difficulty, a set of state transitions is defined for each component within the model of the physical system, with the states being represented by model fragments and the belief in a model fragment imitating the probability that that particular fragment is in the system model. Using the same example there will be four matrices, each of which is of size 5×5, thus, the complexity only increases linearly with respect to the number of components.

The Bayesian approach could have been used to perform the belief revision [22], however, the difficulty in calculating (or approximating) the necessary probabilities, makes their use problematic for the present problem. In particular, there would be no specific evidence to indicate what dependencies (if any) exist between the fragments of various components.

Spohn's work on dynamic epistemic states [23] proposes a method for state transitions based upon positive and negative evidence. This work, rather than concentrating on actual beliefs in states, records relative ordering of these states (with respect to belief). Any penalizing, or confirming, evidence is used to move the whole set of affected states down, or up, this ranking. The result is a mechanism for recording the relative beliefs in states. The difficulty with this approach is that it requires a relatively more complex representation than the current work, as the states are not represented by a single ordering, rather by a series of related orderings (each of which contains a subset of states). The overall ordering is therefore not explicit.

Rewarding Evidence: To achieve a reasonable reassignment of belief, several criteria must be satisfied. First of all, for any component, one of the fragments must be correct (since unknown fault behaviors are explicitly represented by the unknown fragment). Thus, the sum of the revised beliefs *must* remain to be 1 [9] to satisfy the requirement in Markov chains. To ensure this, it is sufficient that the values in each of the columns in the Markov matrix sum to 1. Rewarding evidence suggests that the fragment appears to represent the observed behavior, consequently, the belief in that fragment should be increased. To preserve the Markov properties, the beliefs in some (or all) of the other fragments must be reduced. If the relative ordering of the other fragments is to be retained (as they are independent there is no evidence for or against any of them), the belief acquired from each fragment should be directly proportional to the belief in that fragment. The redistribution of belief is proportional to the current belief to ensure that the relative beliefs in the other fragments is unchanged. This is due to the independence of these fragments and the lack of any evidence to suggest that any of them are more (or less) likely to represent the actual fault. This approach is, essentially, the same as used by Spohn [23].

For the present application given rewarding evidence $E_i \in (0, 1]$, the matrix constructed using elements as defined below satisfies the identified criteria, where a_{xy} is the element in the xth row and yth column of the constructed matrix

$$\begin{aligned} a_{\mathrm{ml}} &= 0, \ \forall\, l, \quad m \in \{1, \ldots, n\}, \quad l \neq m, \quad m \neq i \\ a_{ii} &= 1 \\ a_{il} &= E_i, \quad \forall\, l \in \{1, \ldots, i-1, i+1, \ldots, n\} \\ a_{ll} &= 1 - E_i, \quad \forall\, l \in \{1, \ldots, i-1, i+1, \ldots, n\}. \end{aligned}$$

Penalizing Evidence: As with the case of having a piece of rewarding evidence, several criteria must hold to enable a reasonable reassignment of belief if penalizing evidence is given. The sum of the revised beliefs should also remain as 1 and so each of the columns of this matrix must also sum to 1. To reflect the size of the evidence, the belief in the penalized fragment should be reduced in direct proportion to the size of the evidence. If the belief were assigned to the unknown behavior category, the belief in this unknown behavior would (for those components which are under suspicion) very quickly become the most believed fragment. This would result in the unknown behavior being suggested as part of a fault candidate. As it is not possible to refute (or confirm) an unknown behavior, this fragment would remain the most believed. The known faults, which may be far more likely than an unknown fault would therefore be overlooked due to the relatively high belief in the unknown behavior.

The elements of a Markov matrix that satisfies all these criteria are defined as follows, for given evidence $E_i \in [-1, 0)$, where c_j is the current belief in fragment j:

$$\begin{aligned} a_{\mathrm{ml}} &= 0, \qquad \forall\, l, \quad m \in \{1, \ldots, n\},\ l \neq m,\ l \neq i \\ a_{ii} &= 1 + E_i \\ a_{\mathrm{mi}} &= -E_i \times \frac{c_m}{1 - c_j} \\ &\forall\, m \in \{1, \ldots, i-1, i+1, \ldots, n\} \\ a_{\mathrm{mm}} &= 1, \quad \forall\, m \in \{1, \ldots, i-1, i+1, \ldots, n\}. \end{aligned}$$

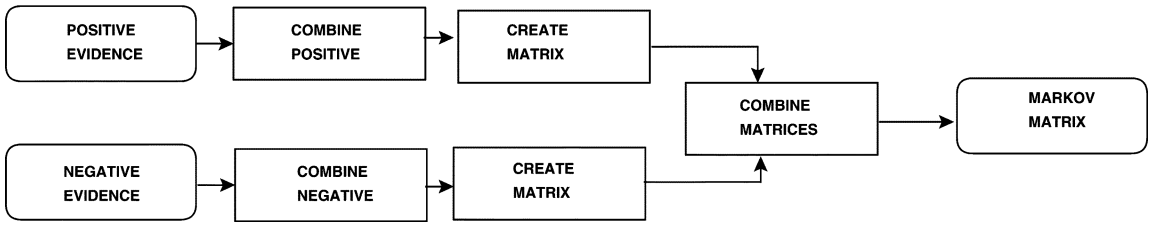

Fig. 2. Combining pieces of evidence.

Combining Evidence: As fragments can belong in more than one conflict or confirm set, there will be more than one piece of evidence relating to a given model fragment. Thus, the Markov matrices derived from individual pieces of evidence need to be combined to form an overall matrix.

The combination of evidence must also satisfy certain basic intuitive criteria. Firstly, two pieces of rewarding evidence must reinforce each other so that the combined evidence is greater than either of the two individual pieces. Similarly, two pieces of penalizing evidence must also reinforce each other, again, so that the overall evidence is greater than either of the original pieces. The combination mechanism should be associative, so that the order of combination of the evidence is not important. An existing technique that satisfies all of these criteria is the Dempster–Shafer theory of evidence (DST) [24]. Owing to its sound mathematical foundations, DST is employed here to calculate the belief in combined evidence. DST generates two pieces of evidence (positive and negative) which need to be combined. The simplest way to do this is to integrate the two matrices by averaging their corresponding elements, which is implemented herein. An alternative approach would be to multiply the two matrices together, however, this is not only computationally more complex but the result is dependant on the order of combination. The steps used in this process are shown in Fig. 2.

This two-stage process is required to allow several pieces of positive (or negative) evidence to reinforce each other. This reflects the increase in the likelihood that a particular component is faulty as the more conflict sets a component is in, the more likely it is to be faulty. The second stage allows any components that have both positive and negative evidence against them have the overall size of the evidence reduced, reflecting the conflicting evidence. Neither of these two steps would be sufficient on their own. If each piece of evidence were used to create a Markov matrix and the resulting matrices were combined by averaging over all of the matrices, the desired properties would not be displayed. The resulting matrix would be a Markov matrix, but the effect of the matrix would not reflect the individual pieces of evidence. If all of the evidence, both positive and negative, were to be combined together before creating a Markov matrix a different problem would arise. The DST only works for positive evidence. The DST could be modified to have two propositions, however, this would result in two pieces of evidence which would still need to be combined. The problem could be avoided by using the method for combining conventional certainty factors (CFs) [25], however, this method is not associative and so the result of combining beliefs would depend upon the order of their combination.

A simplification to the DST has been developed [26], that simplifies some of the computational complexities associated with the DST. In particular, the work described in that paper reduces the complexity, by only considering singletons (sets that only contain a single element) rather than the full-sets allowed on DST. This use of singletons reduces the complexity significantly as the number of such singletons grows linearly with respect to the number of hypothesis. Another feature of this work that is of particular relevance to the present research is the combination of positive and negative evidence (relating to a single hypothesis). Its modification of DST allows for the combination of positive and negative evidence, but still results in two pieces of evidence, which still need to be combined.

In the present work, there are only two possible elements, F and $\overline{F}$ (where F represents the case of the fragment being in the system model and $\overline{F}$ represents the case of the fragment not being in the model), and only four mutually exclusive hypotheses are possible, $m(\{F,\overline{F}\})$, $m(\{\})$, $m(\{F\})$, and $m(\{\overline{F}\})$. This restriction on the values used by DST greatly simplifies the process and results in only two values being stored. The modified DST is in an even simpler version than proposed in [26], as the number of hypothesis is fixed and so complexity is not a major issue, indeed as only beliefs of a single sign are combined the version of DST used in this paper is more efficient.

IV. Guiding Candidate Generation With Beliefs

Having presented the method for belief updating, this section describes a framework (Fig. 3) that utilizes the work to develop a diagnostic algorithm for fault identification. The diagnostic process involves several steps as detailed below:

1) Initialize beliefs;
2) Select a system model;
3) Detect conflicts in the model;
4) If conflicts exist, then:
   a) Generate evidence and evaluate its size;
   b) Update beliefs;
   c) Go to Step 2;
5) Else, report candidate found.

A. Initialize Beliefs

The beliefs in each of the model fragments are initially set to their default values, which are set to their prior probabilities if known. The search for fault candidates causes the initial beliefs to be repeatedly revised. As a result, when a fault candidate is found, the beliefs can be considerably different from their initial values. These new values are a reflection of all of the models

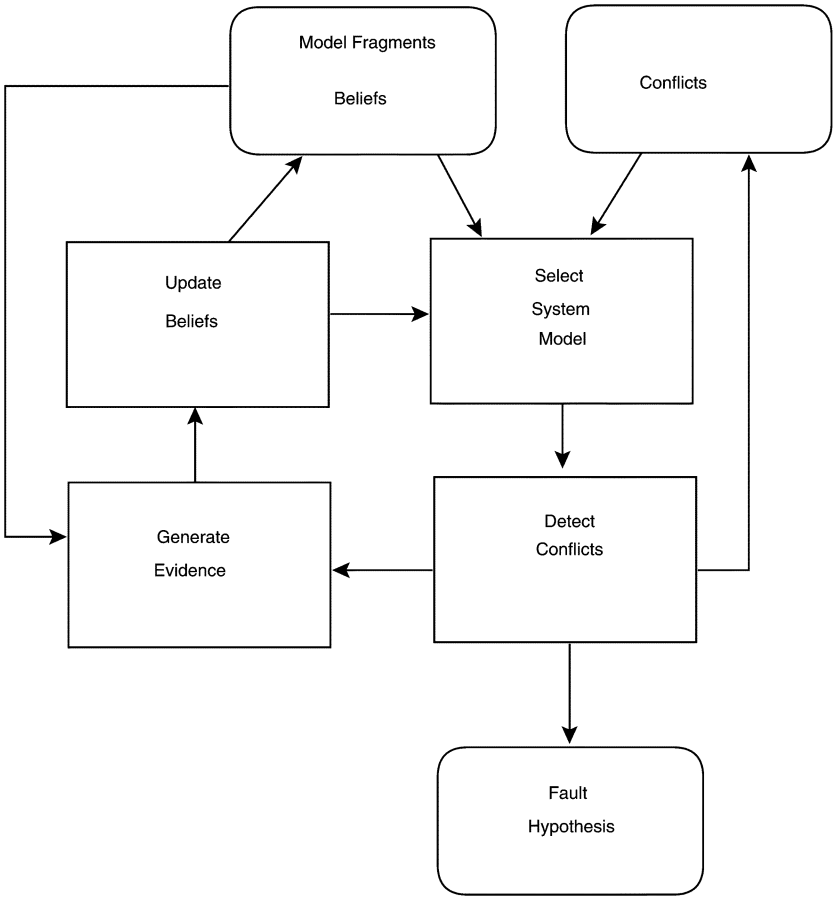

Fig. 3. Key stages of the algorithm.

that were considered during the search for the previous fault candidate. If the search for the next fault candidate were to start from these revised beliefs, the search would be focused on the same candidates which would make it less likely that the next most believed model (in terms of prior probability) would be selected. If the beliefs were reset to their initial values, the search would not be biased by the results of the previous search and so the most believed model is more likely to be found.

B. Select a System Model

The next step involves selecting a system model that consists of one model fragment for each system component. This is done by choosing the model fragments that have the highest belief, ensuring that the most believed model is selected. If there exist more than one model fragment that are of the highest belief for a particular component, one of such fragments is, effectively, picked at random (as there is no justification for choosing one over the other). In order to prevent a model that contains known inconsistencies being selected, an ATMS is used to record the minimal conflicts that are generated during each conflict recognition step, the information in the ATMS is then used to check potential models as they are developed, thus disallowing models with a subset of model fragments that are known to be in conflict. In this way the most likely model that contains no known inconsistencies is chosen.

The process for selecting the next model is a search through the space of models looking for the most believed model that is consistent with the known conflicts. The approach taken is to perform a depth first search through the model space (any given heuristics may of course be used to increase the search efficiency).

C. Detect Conflicts in the Model

Once a system model has been selected, the individual fragments are treated as the *normal* behavior model fragments in a conventional GDE algorithm. The conflict recognition step of the GDE algorithm is run, and any detected minimal conflicts represent sets of model fragments that cannot coexist in the model that represents the actual behavior of the system under diagnosis. These minimal conflict sets are recorded in the ATMS to prevent any future model being generated that contains them as a subset.

D. If Conflicts Exist

If the previously simulated model generated conflicts a candidate has not been found and so the current beliefs should be revised to facilitate the selection of the next model.

1) Generate Evidence and Evaluate Its Size: In general, application of GDE's conflict recognition method will lead to both penalizing and rewarding evidence being found. The penalizing evidence is based upon the minimal conflict sets detected at the last step. A minimal conflict set must, by definition, contain at least one model fragment that does not represent the actual behavior and therefore should not be in the required model. There

is no explicit information about which member of the set causes the conflict, however, the current belief in each of the fragments is known and so the most suspect fragments are those with the lowest belief.

As at least one fragment must be pushed out of the required system model, so that the conflicts generated cannot recur, a total penalizing evidence of 1 can be attached to each of the minimal conflict sets. In the extreme case where the conflict set contains a single value, this value ensures that the fragments belief is reduced to zero. This penalizing evidence is then distributed amongst the model fragments within it in proportion to $1 - B_i$, where B_i is the belief in fragment i. The penalizing evidence E_i is therefore defined as: $E_i = -(1 - B_i/\sum_{j=1}^{n}(1 - B_j))$ where n is the number of elements in the conflict set, clearly $E_i \in [-1, 0)$. As a result, the most *disbelieved* model fragments have the most evidence against them. Note that the approach taken in [25] is to assign negative evidence as positive evidence for all of the other behaviors. Following this, the belief would be equally distributed amongst all of the other fragments. The reason that this approach was not chosen was due to the unknown behavior fragment. If the belief was equally distributed, then the unknown behavior would also get an equal share of the belief. This may not seem unreasonable, but as the unknown fragment can never be disproved (it is impossible to show that a component is not behaving in an unknown way) the belief in the unknown fragment cannot be reduced. As a result, if the belief was distributed evenly, then the unknown fragment would quickly become the most believed and suggested as a (or part of a) possible fault. This description relates to the generation of the evidence, whilst the description of the section Rewarding Evidence refers to the use of this evidence to revise the beliefs.

The rewarding evidence is generated using minimal confirm sets. From the same observed values, finding two ways to predict the same value for a variable suggests that the model fragments upon which the predictions were based are correctly modeling the observed behavior. It is therefore reasonable to increase the belief in these model fragments so that they are more likely to be selected to be part of future models.

If the minimal confirm set only has one element, then that element must be in the model that would explain the observed behavior and so its belief should be increased to 1. If the set contains more than one element, the evidence should be shared amongst each of the elements. The evidence is shared in proportion to the values of the current belief in each of the fragments so that the most believed fragments are allocated most of the evidence, and so

$$E_i = \frac{B_i}{\sum_{j=1}^{n} B_j}, \qquad E_i \in (0, 1].$$

2) Update Beliefs: The final step is to update beliefs in each of the model fragments that have evidence relating to them, using the techniques, based upon Markov chains, described in Section IV-C. If there is more than one piece of evidence relating to a particular model fragment, the pieces of evidence must be combined together. The fault-identification process then selects another system model and continues until a model is found that generates no conflict sets. This model is then returned as a fault hypothesis.

3) Go to Step 2: Now that the beliefs have been revised, the next model to be simulated is selected.

E. Else, Report Candidate Found

As no conflicts were detected in the current model, it is returned as a fault candidate.

V. Comparison With Existing Static Techniques

To illustrate the effectiveness of the techniques, a simple problem case will be considered (complex application examples are to be presented later). The problem is the same as that used in Section II when describing the GDE algorithm [7]. The GDE algorithm is not considered here mainly because GDE only considers a component to be faulty or not faulty. The comparison will be between systems that incorporate fault models instead. The results highlight the differences between the approaches and reflect the relative computational complexity of each of them.

All of the approaches compared in this section use the same underlying technique for detecting faults (the GDE inference engine for fault candidate generation), indeed all model-based static diagnostic systems share the same core as do some recent approaches to dynamic diagnosis [16], [27]. The discussion of the complexity of each of the techniques will therefore focus on the processes for belief revision. There are two potential sources of increased complexity when scaling this problem to a more realistic one.

- An increase in the number of components. This is the most obvious complexity issue, as the number of components increases, so does the complexity of the belief-revision process.
- An increase in the average number of possible behaviors for each of the components. The relative importance of the number of components is shown by the fact that the number of components will generally be significantly greater than the average number of behaviors. A physical system may typically have several thousand components and yet less than ten behaviors per component.

A. System Under Diagnosis

The example used here is not complex, however, the effects of scaling the problem up to a more realistic size are addressed. The illustrative system has been shown in Fig. 1.

B. Markov Chains for Belief Revision

The first technique that will be considered is the work proposed in this paper, the use of Markov chains for belief revision. Suppose that each of the components in the system are modeled by seven different model fragments and, for simplicity, that all of the components have the same set of model fragments. In particular, each component has a model fragment that corresponds to the desired "normal" behavior (F_1), five known fault behaviors (F_2 to F_6) and an unknown fault behavior (F_7).

The initial values of the beliefs were set to 0.99 for each "normal" behavior fragment C_{F1}, 0.000 01 for each unknown fragment C_{F7} and 0.001 998 for all of the other fragments C_{Fi}, $i = 2, \ldots, 6$. These value for the "normal" behavior fragments

TABLE I
EVIDENCE GENERATED

Model Fragment	Evidence	Evidence	Combined Evidence
$M1_{F_1}$	$-\frac{1}{3}$	$-\frac{1}{4}$	$-\frac{1}{2}$
$M2_{F_1}$	$-\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{2}{9}$
$M3_{F_1}$	$-\frac{1}{4}$	$-\frac{1}{3}$	$-\frac{1}{2}$
$A1_{F_1}$	$-\frac{1}{3}$	$-\frac{1}{4}$	$-\frac{1}{2}$
$A2_{F_1}$	$-\frac{1}{4}$	$-\frac{1}{3}$	$-\frac{1}{2}$

TABLE II
REVISED BELIEFS

Fragment Name	M1	M2	M3	A1	A2
F_1	0.495	0.44	0.495	0.495	0.495
F_2	0.1	0.111	0.1	0.1	0.1
F_3	0.1	0.111	0.1	0.1	0.1
F_4	0.1	0.111	0.1	0.1	0.1
F_5	0.1	0.111	0.1	0.1	0.1
F_6	0.1	0.111	0.1	0.1	0.1
F_7	0.0005	0.0006	0.0005	0.0005	0.0005

is reasonable as in general most components will not fail and so the belief in each of them should be relatively high. Similarly, the belief in the unknown fragment is low as it is relatively unlikely to fail in an unknown manner.

When the diagnostic process is invoked, there are no known conflict sets yet and the most believed system model is the "normal" behavior model. This model is selected and analyzed using GDE's conflict recognizer. The following minimal conflict sets were generated: $\{M1_{F_1}\ A1_{F_1}\ M2_{F_1}\}$, $\{M1_{F_1}\ M3_{F_1}\ A2_{F_1}\ A1_{F_1}\}$, and $\{M2_{F_1}\ M3_{F_1}\ A2_{F_1}\}$, resulting in pieces of penalizing evidence. Table I shows the evidence generated by this system model.

The values of the beliefs are then revised, using the penalizing evidence against each of the fragments in the system model. Table II shows the revised values of belief in each of the fragments. From these revised beliefs and the known conflicts, the most likely model, which contains no known conflicts is selected and the process iterates. In this case the selected model contains the following fragments: $M1_{F_1}$, $M2_{F_2}$, $M3_{F_1}$, $A1_{F_2}$ and $A2_{F_1}$.

After seven iterations, a fault hypothesis is found with $M1_{F_2}$ and $M2_{F_5}$ being the faulty components, which correctly explains the observed behavior. The fragment $M2_{F_5}$ was the fifth model fragment that had been considered for component $M2$, and so, most of the system models that were considered were there to evaluate the various possibilities for component $M2$. If model fragment $M2_{F_5}$ had represented the most likely fault behavior for component $M2$, then the fault would have been identified after only three iterations, as the most likely fragment would have been considered first.

The use of conflict sets helps to significantly reduce the number of combinations that needs to be considered. So, for example, if components $M1$, $M2$, and $A2$ had become the focus of the search, then it would become a problem of identifying the combination of fragments of these three components that might explain the behavior. As there are five possible behaviors for each of these components, there are 125 possible combinations. By using the conflict sets, this number can be significantly reduced, by omitting those areas of the search space that contain a conflict set as a subset.

Each of the components has its belief revised independently of the others, though the values of the beliefs are dependant on the behavior of the other components. Additionally the cost of revising the belief for a component is fixed (assuming that the number of possible behaviors stays constant). Thus, the effect of doubling the number of components is to double the effort involved in revising the beliefs. The complexity of the belief revision process is thus linear with respect to the number of components in the model.

The average number of possible behaviors on the other hand has potentially a greater complexity. The generation of the Markov matrices used is itself linear as the matrices are sparse, as each matrix contains $3n-2$ nonzero elements, with n being the average number of behaviors. The complexity increases for the application of the matrices as the process is effectively a matrix multiplication and so rather than being linear the complexity is $O(n^2)$. This is not a serious problem as long as n is relatively small.

C. GDE+

An existing system that utilizes a similar set of possible fault behaviors is GDE+ [17]. It starts in the same way as in the previous example and generates the same minimal conflict sets. The process then determines the minimal fault candidates (as in GDE), resulting in the following: $\{M1,\ M2\}$, $\{M1,\ M3\}$, $\{M1,\ A2\}$, $\{A1,\ M2\}$, $\{A1,\ M3\}$, $\{A1,\ A2\}$, $\{M2,\ M3\}$, $\{M2,\ A2\}$. GDE+ now tries to evaluate possible faults from all the possible fault behaviors. If it is assumed that all of the possible behaviors are as given in the previous example there are five possible faults for each component (if the unknown behavior is ignored as GDE+ simply cannot handle this). There are therefore 25 possible fault combinations for each of the minimal candidate sets. As there are eight minimal candidate sets, a total of 200 fault combinations need to be considered (assuming that only two components are actually faulty). GDE+ is therefore particularly affected by an increase in the average number of possible behaviors.

When GDE+ is applied to larger, more practical, diagnostic problems, the complexity greatly increases. As the number of components increases, so generally does the size of the conflict sets, this is because the predicted values tend to be derived through more components. Additionally, if the system under diagnosis has more inputs and outputs, the number of conflict sets also increases. While the computational complexity of GDE+ is not directly dependant on the number of components in the system under diagnosis, this increase in the size of conflict sets has a significant effect. As the size or number of conflict sets increases, the number of minimal fault candidates increases and therefore the number of fault combinations that GDE+ needs to consider also increases.

The relative number of fault combinations that needs to be considered depends upon the number of faulty components in the system under diagnosis. If there is only a single fault, the number of fault combinations only increases linearly. This is because each additional member of a conflict set only requires a fixed increase (the number of possible faults) in the number of fault combinations. However, when two or more faulty components are present the complexity increases exponentially.

D. Sherlock

Another system that uses an identical type of possible model (including the unknown fault model) to the present work is Sherlock [18]. Sherlock starts in the same way as GDE+. The minimal conflict sets are used to focus the search for possible diagnoses. In order to generate leading candidates it then uses approximate prior probabilities in the same form as the Markov chain process. It only considers the most likely faults initially, and only considers the less likely faults if the most likely ones have been eliminated. The problem with this approach is that, in order to find leading candidates, a considerable number of potential candidates may have to be considered.

The above problem is compounded if the size and number of the conflict sets increases, as this significantly increases the number of possible models that need to be considered. The process is roughly linear with respect to the number of components primarily due to the focusing mechanism. The complexity is more dependant on the size and number of the conflict sets, as either of these increases so generally does the size of the focus of the diagnosis, which subsequently increases the complexity. The effect is similar to that observed in GDE+, except that as Sherlock searches for the most likely faults, the complexity with respect to the average number of behaviors is partially dependant on the relative probabilities of each of the behaviors. However, as Sherlock may consider candidates whose probability is 1/100th of the best candidate, an increase in the number of behaviors can have a considerable detrimental effect on the candidate selection process.

E. Summary

The process that uses Markov chains for belief revision has complexity advantages over both GDE+ and Sherlock, both in terms of the number of components and the average number of behaviors. In particular it uses information from each model simulation to guide successive candidate selection processes. In addition, as it only considers one candidate at a time, the potential for complexity problems is considerably reduced.

The work presented in this paper could also be compared to the existing static techniques in relation to the order in which possible candidates are considered. All of the existing techniques are guaranteed to find the most believed candidates first, as they consider candidates in strictly decreasing order of belief. The work presented here may not necessarily consider the candidates in such a strict order. The actual effectiveness of the current work will be discussed further in the results section.

VI. Experimental Results

To demonstrate the utility of the present work, two sets of results are presented. The first set of results systematically tests a nontrivial problem, in the second set of results, the present work is applied to a significantly complex problem of identifying faults.

A. Systematic Experimental Evaluation of the Approach

This section presents the results of systematically applying the techniques described in this paper to a nontrivial diagnostic problem. Multiple faults will be simulated in the system under diagnosis and then the performance of the diagnostic process will be evaluated by its ability to identify the simulated faults in as few attempts as possible.

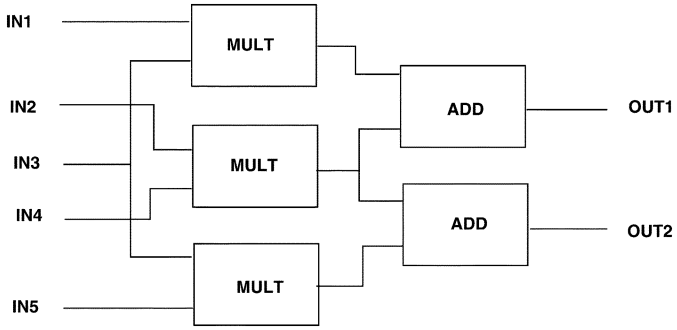

Fig. 4. Full adder.

1) System Under Diagnosis: The system under diagnosis is built from full adder modules with one such module shown in Fig. 4. These modules each consist of five individual components (three multipliers and two adders) and have five inputs and two outputs. Overall, there are fifty inputs ($A1..A10$, $B1..B10$, $C1..C10$, $D1 \ldots D10$, and $E1 \ldots E10$) and 16 outputs ($o1 \ldots o16$) in the system. The system has a total of 90 components (54 multipliers and 36 adders) and each of the inputs has an effect on up to eight of the outputs. The system is therefore reasonably complex and provides a test bed for evaluating the techniques described in this paper.

Behavioral Fragments for Each Component: For each of the components in the system model four possible model fragments were defined, each of which could explain the behavior of the component:

1) Normal behavior. The component behaves as it is expected to, representing the component not being faulty. The prior probabilities, $p(\text{normal})$, were set to 0.99 to reflect the low likelihood that any individual component is faulty.
2) Stuck at zero (denoted by s0). This fragment represents the fact that the component returns the value zero, no matter what the value of the inputs. The prior probabilities, $p(s0)$, were set to 0.004 999 9 as it was assumed that known faults were more likely than unknown faults.
3) Stuck at one (denoted by s1). This time the output is 1 no matter what the inputs are. The prior probabilities, $p(s1)$, were set to the same value as $p(s0)$, 0.004 999 9.
4) Unknown (denoted by u). This fragment represents the case where the fault is unknown. The prior probabilities, $p(u)$, were set to 0.000 000 2, as it was assumed that an unknown fault was unlikely.

The model used has 90 components of various types, each of which has four possible behaviors. The total number of possible models is therefore $4^{90} \approx 1 \times 10^{54}$. Clearly it would be impossible to try all possible models and find possible faults exhaustively.

2) Experimental Methodology: To evaluate the diagnostic method described here, it is interesting to investigate its performance over a significant number of diagnoses. A total of 100 multiple faults were simulated for this purpose. Each of the attempts at diagnosis was generated randomly. In each case, two

TABLE III
NUMBER OF MODELS SIMULATED FOR EACH TEST

Test	Models	Test	Models	Test	Models	Test	Models
1	3	26	8	51	2	76	14*
2	3	27	4	52	2	77	2
3	2*	28	13*	53	2	78	3
4	7*	29	4	54	11*	79	2
5	11*	30	2*	55	4	80	4
6	6*	31	1	56	2	81	1*
7	2	32	7*	57	1	82	9*
8	4	33	3*	58	4	83	7
9	3*	34	5	59	1*	84	4
10	15*	35	11	60	8*	85	1
11	3	36	4	61	3	86	1
12	7	37	6	62	4	87	10*
13	10*	38	3*	63	2	88	2
14	4*	39	4	64	3	89	4*
15	4	40	2	65	11*	90	2
16	1	41	1*	66	10*	91	2*
17	2	42	4	67	2	92	4
18	7*	43	4*	68	1	93	1
19	3	44	9*	69	6	94	12*
20	3*	45	1*	70	7*	95	2
21	4	46	3	71	2	96	4
22	3	47	2	72	3	97	4
23	4	48	9	73	11*	98	4*
24	2*	49	6	74	1	99	2
25	2	50	5	75	5	100	17*

faults were simulated by randomly selecting two of the components and then randomly allocating them one of the two known faults ($s0$ or $s1$). The inputs for each test were also generated randomly to avoid any bias introduced by a single set of inputs being used in each test. Each test ran until the actual fault was found, if other fault candidates were proposed before the actual fault the test continued.

To evaluate the performance of each test, two factors were measured. First, the number of models simulated before the correct faults were found. This measurement is important, as the aim of the diagnostic process is to find the actual fault as quickly as possible. The second method is to consider the prior probability of each of the models that were simulated. It is important that the models should be considered in order of prior probability, otherwise, less likely candidates may be suggested before more likely ones.

3) Results: The results are presented in this section. The tables that show the results of individual tests only indicate those components that are not behaving in their nonfault mode.

Number of Models Simulated: The number of models simulated for each test are summarized in Table III. The average number of models simulated in these one hundred tests was 4.62, indicating that the process is very efficient at identifying faults in this nontrivial physical system. In 11 cases, the correct fault was identified in the first model to be simulated. There were thirteen tests that took ten or more models, with the largest number of models (17) being simulated in the final test. This compares very favorably against the total number of models that may have to be simulated in exhaustive search. The distribution of the number of models simulated is shown in Fig. 5. This demonstrates that the majority of the faults are correctly identified within four models, and only a few tests taking more than 11 models to correctly identify the faults.

The tests that took four or less model simulations had generally to do with the cases where the faults did not interact with each other. These cases were effectively two single faults and the search for fault candidates was a search for two distinct faults, which quickly focuses on a small set of components. An example of a test where the correct faults were detected very quickly is test 8, the details of which are shown in Table IV. In this case, one of the faulty behaviors is correctly identified in the first model simulated (component add13 displaying fault $s0$), the subsequent models are then used to search for the second fault. Fig. 6 shows that most of this kind of fault were correctly identified within four models, and all of such faults were correctly identified within nine models. The generally low number of models considered in these tests confirm that such tests are simpler to diagnose.

The tests that took more than ten model simulations were generally tests where the faults did interact with each other. In these cases, the search for fault candidates is more difficult, as a model that contains only one of the faults does not reduce the number of conflict sets. These tests are indicated in Table III by having a $*$ in the Models column. An example of a test which needs more than ten model simulations is test 5, the details of which are shown in Table V. In this example, one of the actual faults (component *mult 31* displaying fault $s1$) is found after only four simulations, with the remaining simulations trying to find the other fault. These results are summarized in Fig. 7, which shows that most of this kind of fault were correctly identified within eleven models. All of the faults were correctly identified within seventeen models. These tests generally require more model simulations than those where the faults do not interact, which confirms the expected increase in complexity in diagnosing such faults. Despite the generally poorer performance of tests where the faults did interact, only considering an average of approximately seven models is impressive when compared to the number of possible models.

Single Fault Solutions: Several tests generated candidates with only one single fault, despite two faults actually being present. In each of theses cases, the single fault was one of the actual faults that had been simulated. There are two reasons why just one single fault can be suggested when two faults are known to exist.

- A faulty component is not giving an erroneous output. If this occurs, the component is not observed to be faulty.
- The two faults directly interact with each other. In this case, only one of the components will appear to be faulty.

As the diagnostic process only finds minimal fault candidates, it cannot suggest the actual faults as a fault candidate in such cases, though it will identify one of the components as being faulty on its own. Missing one of the underlying faults is therefore acceptable for such difficult situations as the missed one is equivalently subsumed by the identified fault.

Prior Certainties of Simulated Models: In most of the one hundred tests performed, the models were simulated in order of decreasing prior probability. In only two tests were models of a lower prior certainty considered before models of a higher prior probability, namely test 37 and test 100. The results of one of these tests is shown in Table VI.

In test 37, one of the actual faults was found in the first simulated model and was in all of the subsequent models. The other

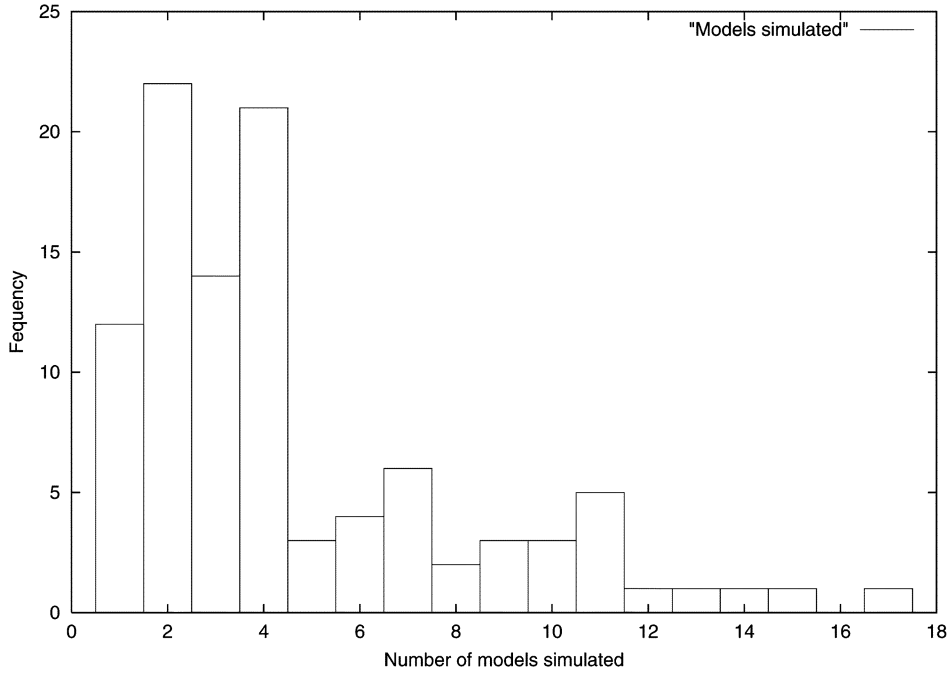

Fig. 5. Summarized results.

TABLE IV
RESULTS OF TEST 8

Model Number	Faulty Component	Behaviour	faulty Component	Behaviour	Prior Certainty
1	*add 13*	s0	*add 33*	s0	1.03E-05
2	*add 13*	s0	*mult 49*	s0	1.03E-05
3	*add 13*	s0	*add 33*	s1	1.03E-05
4	*add 13*	s0	*mult 49*	s1	1.03E-05

faulty component was correctly identified in model 6, however, a different model fragment was considered for this component in model 2.

4) Discussion: The results presented in this section are generally very encouraging, the correct faults are found, on average, in less than five models. The diagnostic process very quickly focuses the search for faults on a few components, with the slowest test taking 17 models to correctly identify the actual faults.

In 98% of the tests, the models were simulated in decreasing order of prior probability. In the other 2% of tests, this ordering did not occur. The reason that these two tests did not follow the ordering was that the beliefs had been revised to the extent that the beliefs in some of fault behaviors rose above that of non-fault behaviors and so models with three faults were considered before those with only two faults. The prior probability in the normal behavior fragment in each component was set to 0.99, combining the certainties for the normal behavior model gave an overall prior probability of 0.405. This suggests that 60% of the time at least one of the components will be faulty. If the prior probability in each of the normal behavior fragments were increased to 0.9999, the overall prior certainty becomes 0.99, suggesting only a 1% chance of faults occurring.

The two tests that did not follow the decreasing prior certainty order were repeated with the prior probability in the normal behavior fragments increased to 0.9999 (the other prior certainties were revised accordingly). The results of one of these additional tests is shown in Table VII. The problem of not selecting models in decreasing order of prior probability has been resolved for both of these cases.

These results show that even with the initial low prior probabilities in the normal behavior fragments, the technique was 98% successful in considering candidates in a decreasing order of belief. Increasing the prior probabilites resulted in this being increased to a success rate of 100%. While there is no guarantee that the technique will always achieve 100% these results indicate that the success rate can be very high.

B. Diagnosing Faults in an ISCAS'85 System

The domain under investigation will be described first, highlighting the reasons for choosing such a system in demonstrating the utility of the present work. Results from various tests will then be presented, and the use of the techniques on such large systems discussed.

1) System Under Diagnosis: The system under diagnosis was taken from a standard test suite of complex systems [28]. The circuit chosen was c1355, which contained 546 components with 41 inputs and 32 terminals (labeled 1324gat to 1355gat). All of the variables take binary values.

Components in the Model: The model used here is built from a range of components as summarized in Table VIII, including and gates with two, four, and five inputs. For each of the components in the system model, the same four possible model fragments were defined, each of which could explain the behavior of the component. Given that the system used for demonstration has 546 components of various types, and each of which has four possible behaviors, the total number of possible models is, therefore, $4^{546} \approx 2 \times 10^{330}$. Clearly, it would be impossible to try all possible models and find possible faults exhaustively.

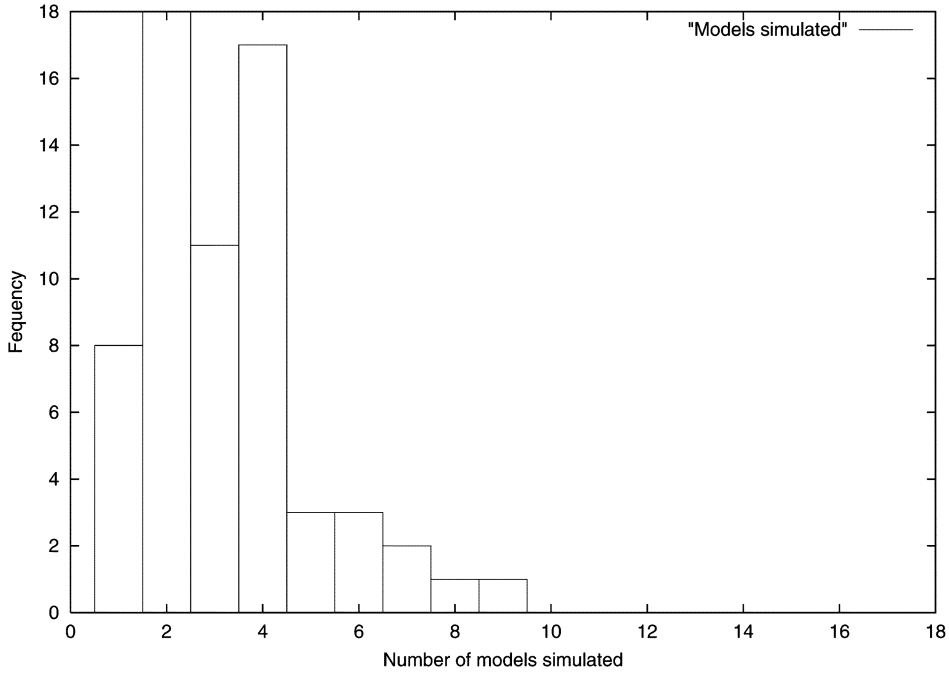

Fig. 6. Summarized results where the faults do not interact.

TABLE V
RESULTS OF TEST 5

Model Number	Faulty Component	Behaviour	faulty Component	Behaviour	Prior Certainty
1	*add 1*	s0	*add 10*	s0	1.03E-05
2	*add 21*	s0	*add 24*	s0	1.03E-05
3	*add 21*	s1	*add 24*	s1	1.03E-05
4	*mult 31*	s0	*mult 36*	s0	1.03E-05
5	*mult 31*	s1	*mult 36*	s1	1.03E-05
6	*add 10*	s1	*mult 31*	s1	1.03E-05
7	*mult 31*	s1	*mult 35*	s0	1.03E-05
8	*mult 14*	s0	*mult 31*	s1	1.03E-05
9	*mult 15*	s0	*mult 31*	s1	1.03E-05
10	*mult 31*	s1	*mult 35*	s1	1.03E-05
11	*mult 15*	s1	*mult 31*	s1	1.03E-05

2) Diagnostic Performance: To demonstrate the effectiveness of the diagnostic program on this sophisticated system, two separate test cases are employed. In each case, the results of several iterations of the diagnostic process will be given to show not only the first fault that it finds, but also the subsequent faults. This is important as there may be several possible ways for the faulty behavior to have been caused, and the first candidate found may not represent the actual faults in the physical system. Thus, allowing for the generation of multiple candidates (each of which may itself involve multiple faults) increases the ability to identify faults correctly.

In both tests, at least two components must be faulty to explain the observed behavior. All of the test cases given here use the same set of inputs; each of the inputs is set to the value 1. If there were no faults in the system, then the expected output values (1324gat to 1355gat) would all have the value 1. To simulate faults in the model, it is therefore only necessary to set one or more of the output values to zero.

First Result: For this example, all of the outputs were set to the value 1, except for 1324gat and 1355gat (the first and last outputs), which were set to the value 0. From observing the structure of the model, it can be seen that it is not possible for a single fault to explain both of these discrepancies and so there must be at least two faulty components. With such an observation, the most obvious solution is that both component *buff 1* and component *buff 32* are faulty.

The first 20 models considered are shown in Table IX, along with the belief in each of the models and an indication as to whether conflicts still exist. Of these 20 models, no fewer than 12 of them are physically meaningful fault candidates. The first model checked happens to be the most obvious candidate for explaining the observed faults, namely *buff 1* and *buff 32* are both behaving as if their output is stuck at the value 1. The use of the conflict sets in guiding the selection of fault candidates is clearly successful as the very first model tried contains two potential faults that jointly explain the observed behavior.

As the two discrepancies were obtained from distinct areas of the system, the pattern that emerges from the results is that the diagnosis is treating each of the discrepancies as the effect of a single fault. This means that the diagnostic process can be interpreted as a combination of two separate diagnostic processes, covering a large number of the models resulting from the combination of these two *single* faults.

Second Result: For this example, all of the outputs were set to the value 1, except for 1354gat and 1355gat (the last two outputs), which were set to the value 0. Again, it is not possible for a single fault to explain both of these discrepancies and so there must be at least two faulty components. The most obvious solution is that both component *buff 31* and component *buff 32* are faulty.

The first 16 models considered are shown in Table X, along with the belief in each of the models and an indication as to whether conflicts still exist. The elements that contain no values for the second Faulty Component and Behavior column, were

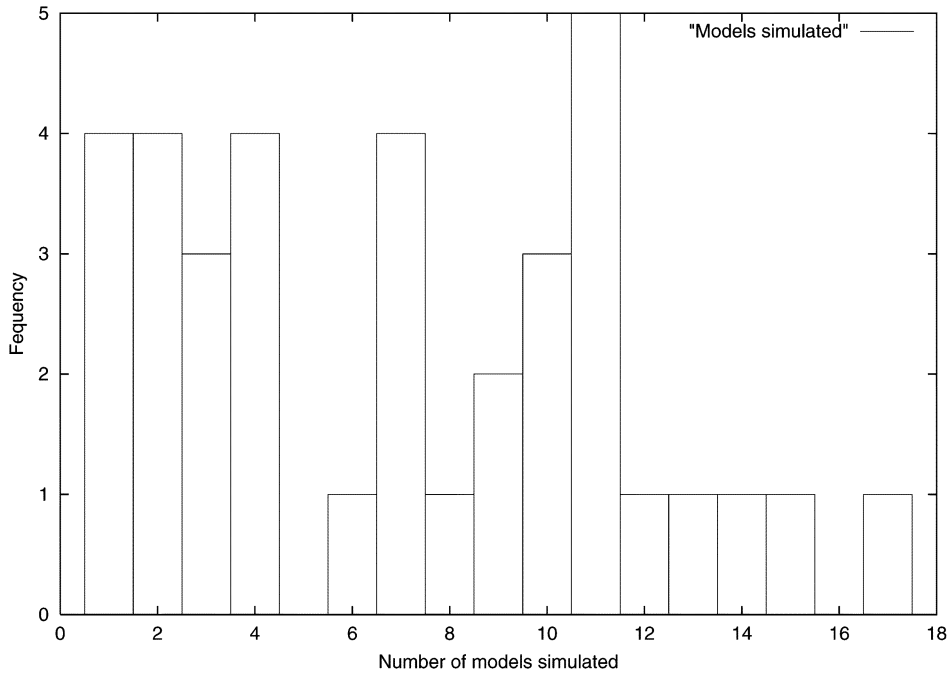

Fig. 7. Summarized results where the faults interact.

TABLE VI
RESULTS OF TEST 37

Model No.	Faulty Comp.	Behav.	Faulty Comp.	Behav.	Faulty Comp.	Behav.	Prior Certainty
1	*add 13*	s0	*add 24*	s0			1.03E-05
2	*mult 19*	s0	*add 24*	s0			1.03E-05
3	*add 13*	s1	*add 24*	s0			1.03E-05
4	*add 15*	s0	*add 24*	s0	*mult 38*	s0	5.21E-08
5	*add 24*	s0	*mult 38*	s1	*mult 47*	s0	5.21E-08
6	*mult 19*	s1	*add 24*	s0			1.03E-05

TABLE VII
RESULTS OF THE REVISED TEST 37

Model Number	Faulty Component	Behaviour	faulty Component	Behaviour	Prior Certainty
1	*add 13*	s0	*add 24*	s0	1.03E-05
2	*mult 19*	s0	*add 24*	s0	1.03E-05
3	*add 13*	s1	*add 24*	s0	1.03E-05
4	*mult 19*	s1	*add 24*	s0	1.03E-05

TABLE VIII
COMPONENTS IN THE MODEL

Component Type	Number in model
$andgate$	56
$buffgate$	32
$nandgate$	416
$notgate$	40
$orgate$	2

attempts by the diagnostic program to explain the discrepancies with a single fault. The results follow much the same pattern as in the previous example, using the revised beliefs and conflict sets to generate models until a "partial" solution is found and then focusing on the rest of the model. A total of seven different candidates were found, as summarized in Table X.

Despite the fact that a single fault cannot explain the two discrepancies, there are four models tried that only contain a single

TABLE IX
RESULTS OF THE SECOND TEST

Model Number	Faulty Comp.	Behav.	Faulty Comp.	Behav.	Prior Certainty	Still Conflicts?
1	*buff1*	s0	*buff32*	s0	2.33E-08	NO
2	*and9*	s0	*buff32*	s1	2.33E-08	YES
3	*nand383*	s0	*nand385*	s1	2.33E-08	YES
4	*nand383*	s1	*buff1*	s1	2.33E-08	YES
5	*nand321*	s0	*nand416*	s1	2.33E-08	YES
6	*nand385*	s0	*buff32*	s0	2.33E-08	NO
7	*nand383*	s1	*buff1*	s0	2.33E-08	NO
8	*nand385*	s0	*nand416*	s0	2.33E-08	NO
9	*nand320*	s0	*nand321*	s1	2.33E-08	NO
10	*and40*	s0	*nand289*	s1	2.33E-08	YES
11	*and40*	s1	*buff1*	s1	2.33E-08	NO
12	*nand289*	s0	*nand320*	s1	2.33E-08	YES
13	*nand289*	s0	*buff32*	s0	2.33E-08	NO
14	*nand289*	s0	*nand416*	s0	2.33E-08	NO
15	*nand320*	s0	*buff1*	s0	2.33E-08	NO
16	*nand289*	s0	*nand383*	s1	2.33E-08	NO
17	*nand320*	s0	*nand385*	s0	2.33E-08	NO
18	*5and1*	s0	*nand416*	s0	2.33E-08	YES
19	*5and8*	s0	*nand321*	s1	2.33E-08	YES
20	*nand289*	s0	*nand320*	s0	2.33E-08	NO

fault. The reason for this is that the conflict sets generated allow for components *5 and 8*, *nand 153*, and *nand 154* to singly explain the fault logically. To explain the reason for this, consider a simple model, as shown in Fig. 8.

In this simple system, there are two discrepancies (the outputs should be 1 and 0, not 0 and 1). The conflict recognition phase of GDE would suggest that both of these discrepancies could be explained by a fault in component *buff 1*. However, for *buff 1* to explain both faults, it would need to output the value 0 to *buff 2* and the value 1 to the not gate *not*, which is clearly impossible. The single fault candidates that occur in Table X are due to this phenomenon, as they generate conflicts they are effectively eliminated from further models. The single fault frag-

TABLE X
RESULTS OF THE THIRD TEST

Model Number	Faulty Comp.	Behav.	Faulty Comp.	Behav.	Prior Certainty	Still Conflicts?
1	*5and8*	s0			1.15E-05	YES
2	*nand154*	s1			1.15E-05	YES
3	*nand153*	s1			1.15E-05	YES
4	*buff31*	s0	*buff32*	s0	2.33E-08	NO
5	*nand415*	s0	*buff32*	s1	2.33E-08	YES
6	*5and8*	s1			1.15E-05	YES
7	*nand320*	s0	*nand415*	s0	2.33E-08	NO
8	*nand383*	s0	*nand415*	s1	2.33E-08	YES
9	*nand381*	s0	*nand416*	s1	2.33E-08	YES
10	*nand415*	s0	*buff32*	s0	2.33E-08	NO
11	*nand381*	s1	*nand416*	s0	2.33E-08	NO
12	*nand319*	s0	*nand320*	s1	2.33E-08	YES
13	*nand319*	s0	*nand383*	s1	2.33E-08	NO
14	*and39*	s0	*buff32*	s0	2.33E-08	YES
15	*and39*	s1	*buff32*	s0	2.33E-08	NO
16	*nand319*	s0	*buff32*	s0	2.33E-08	NO

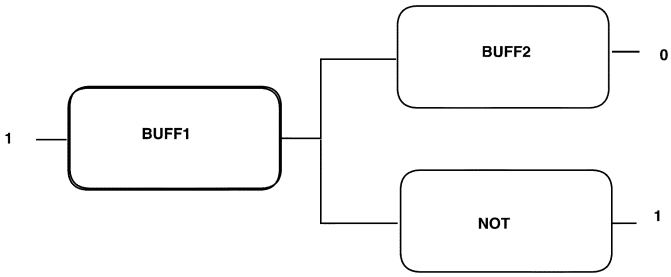

Fig. 8. Two faults reflected though single candidate.

ments will have a relatively low belief, which results in most of the negative evidence being used against them (as the negative evidence is weighted against less believed fragments). As a result, when these single faults are proposed, large negative evidence is applied during the belief revision phase, which greatly reduces their belief and effectively eliminates them from future models.

The reason that the unknown behavior fragment has not appeared as a fault candidate in the above examples is that the initial belief in the unknown behavior fragments is very small. The single fault candidate that represents the unknown behavior is thus not considered before the double faults by the diagnostic program. If the unknown behavior was given the same belief as the know behaviors, then it would be likely to be considered as part of a fault candidate early in the process. Again, as there is no way to confirm (or refute) an unknown behavior this could give prominence to the unknown fault. It would also hinder the known faults being selected as they can occur in conflict sets and have their beliefs reduced (unlike the unknown behavior).

3) Discussion: The results presented in this section clearly show that the combination of belief revision, conflict sets and confirm sets lead to an efficient method of fault candidate generation and identification. The use of conflict sets effectively prunes the search space so that models that can be discarded on the basis of known conflicts are never considered again if any subset of a known conflict has already been refuted.

The results demonstrate that the techniques developed here are readily applicable to larger systems, indeed the complexity of the belief revision process only increases linearly with respect to the number of components. The use of conflict sets enables the model selection process to eliminate large portions of the search space and the use of belief-revision focuses the search even more. The use of confirm sets allows the belief-revision process to increase the beliefs of components that are not in conflict. The combination, of the belief-revision process and the conflict sets, effectively focuses the candidate search.

Another factor that aids the ability of the present diagnostic process to work on scaled up systems is that a larger number of components does not necessarily mean a larger number of simultaneous faults. The relative infrequency of faulty components results in the search process concentrating on specific areas of the search space. The search for more than one faulty components out of 500 components is only slightly more complex than searching for more than one faulty components out of 100. Though the conflict sets themselves may be larger, the minimal candidate sets will still only contain a few components and so the search will focus on these components. The complexity in propagating values and selecting models, however, increases more significantly. Further work is therefore required to improve the model selection process.

This example shows that despite the increase in the size of the system under diagnosis, the complexity increase has been minimized due to the linear nature of the approach.

VII. CONCLUSION

This paper has presented a novel approach for fault identification that integrates Markov chains and GDE-style symbolic conflict recognition. System models are selected based upon the current belief in the behavioral descriptions of its individual components and the results of each model simulation are used to revise these beliefs. This is achieved by converting the evidence, gathered through the conflict recognition process, to Markov matrices which facilitate the belief revision. This combination of Markov chains with GDE-style symbolic conflict recognition, has led to an approach that extends the capabilities of GDE type systems. The combination of belief revision and the use of an ATMS to record conflicts allows for a reduction in the number of possible models that need to be considered. Additionally, further work could be undertaken to extend the approach to diagnose faults in dynamic systems.

The results that have been obtained so far suggest that this approach offers a great potential in performing efficient fault diagnosis. The systematic evaluation of the approach showed that in 98% of the tests the models were simulated in decreasing order of prior probability and that in all cases the actual faults were correctly identified. Using the techniques on a larger system showed the ability of the technique to be scaled up. The main reasons for this are that the belief revision process is linear and that the use of conflict and confirm sets effectively prunes the search space and focuses the search.

Further work is still required, however, for the method proposed to be extended into a fully fledged diagnostic system, rather than just a candidate proposer. The fully fledged diagnostic system would endeavour to distinguish between candidates by suggesting additional measurements or proposing alternative inputs. It would be very interesting to compare the present

work with other most recent approaches (e.g., [29]), which employ alternative uncertainty handling techniques for diagnosis. In addition, the efficiency of the present work, especially the selection of system models, may be further improved by the use of dynamic flexible constraint satisfaction techniques [30].

The current system only records the conflict sets generated at each stage of the diagnostic process. A significant improvement in the performance of the process could be achieved if all of the propagated values within the candidate proposer were stored. This stored information could then be extracted to form the basis of future model simulations. This improvement would be especially significant when only a relatively few components vary between models, as most of the value propagation would already have been performed, thus reducing the amount of calculation required. The increase in efficiency would arise from only having to propagate values through the newly added model fragments.

REFERENCES

[1] L. Trave-Massuyes and R. Milne, "Gas-turbine condition monitoring using qualitative model-based diagnosis," *IEEE Expert*, vol. 12, pp. 22–31, May/June 1997.

[2] M. J. Chantler, G. M. Coghill, Q. Shen, and R. R. Leitch, "Selecting tools and techniques for model-based diagnosis," *Artif. Intell. Eng.*, vol. 12, no. 1, pp. 81–98, 1998.

[3] L. Console, L. Portinale, and D. T. Dupre, "Using compliled knowledge to guide and focus abductive diagnosis," *IEEE Trans. Knowledge Data Eng.*, vol. 8, pp. 690–706, Oct. 1996.

[4] W. Buntine, "A guide to the literature on learning probabilistic networks from data," *IEEE Trans. Knowledge Data Eng.*, vol. 8, pp. 195–210, Apr. 1996.

[5] F. S. Smith and Q. Shen, "Assisting fault identification through markov chains," in *Proc. 10th Int. Workshop Principles Diagnosis*, 1999, pp. 242–249.

[6] ——, "Combining symbolic conflict recognition with markov chains for fault identification," in *Proc. 4th IFAV Symp. Fault Detection Supervision Safety Tech. Processes*, vol. 2, 2000, pp. 1086–1091.

[7] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artif. Intell.*, vol. 32, pp. 97–130, 1987.

[8] J. de Kleer, "An assumption-based tms," *Artif. Intell.*, vol. 28, pp. 127–162, 1986.

[9] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*. Princeton, NJ: Princeton Univ. Press, 1994.

[10] W. Hamscher, L. Console, and J. de Kleer, *Readings in Model-Based Diagnosis*. San Francisco, CA: Morgan Kaufmann, 1992.

[11] B. Falkenhainer and K. D. Forbus, "Compositional modeling: Finding the right model for the job," *Artif. Intell.*, vol. 51, 1991.

[12] J. de Kleer, "Focusing on probable diagnoses," in *Proc. AAAI*, 1991, pp. 842–848.

[13] P. J. F. Lucas, "Analysis of notions of diagnosis," *Artif. Intell.*, vol. 105, pp. 295–343, 1998.

[14] J. Mauss and M. Sachenbacher, "Conflict-driven diagnosis using relational aggregations," in *Proc. 10th Int. Workshop Principles Diagnosis*, 1999, pp. 174–183.

[15] K. de Koning, B. Bredeweg, J. Breuker, and B. Wielinga, "Model-based reasoning about learner behavior," *Artif. Intell.*, vol. 117, pp. 173–229, 2000.

[16] P. Struss, "Fundamentals of model-based diagnosis of dynamic systems," in *Proc. 15th Int. Joint Conf. Artif. Intell.*, 1997, pp. 480–485.

[17] P. Struss and O. Dressler, "Physical negation—Integrating fault models into the general diagnostic engine," in *Proc. 11th Int. Joint Conf. Artif. Intell.*, vol. 2, 1989, pp. 1318–1323.

[18] J. de Kleer and B. C. Williams, "Diagnosis with behavioral modes," in *Proc. 11th Int. Joint Conf. Artif. Intell.*, 1989, pp. 1324–1330.

[19] B. C. Williams and V. Gupta, "Unifying model-based and reactive programming within a model-based executive," in *Proc. 10th Int. Workshop Principles Diagnosis*, 1999, pp. 281–288.

[20] L. Dinca, T. Aldemir, and G. Rizzoni, "Fault detection and identification in dynamic systems with noisy data and parameter/modeling uncertainties," *Reliability Eng. Syst. Safety*, vol. 65, pp. 17–28, 1999.

[21] W. Grossmann and H. Werther, "A stochastic approach to qualitative simulation using markov processes," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 1993, pp. 1530–1535.

[22] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*. San Francisco, CA: Morgan Kaufmann, 1988.

[23] W. Spohn, *Ordinal Conditional Functions: A Dynamic Theory of Epistemic States*. Norwell, MA: Kluwer, 1988, pp. 105–134.

[24] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton Univ. Press, 1976.

[25] B. Buchanan and E. H. Shortliffe, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison-Wesley, 1984.

[26] J. Gordon and E. H. Shortliffe, "A method for managing evidential reasoning in a hierarchical hypothesis space," *Artif. Intell.*, vol. 26, pp. 323–357, 1985.

[27] D. Dvorak and B. Kuipers, "Process monitoring and diagnosis," *IEEE Expert*, vol. 6, pp. 67–74, June 1991.

[28] P. Brglez, F. Pownall, and R. Hum, "Accelerated ATPG and fault grading via testability analysis," in *Proc. Inst. Elect. Electron. Eng. Int. Symp. Circuits Syst.*, 1985, pp. 695–698.

[29] J. Kohlas, B. Anrig, R. Haenni, and P. A. Monney, "Model-based diagnostics and probabilistic assumption-based reasoning," *Artif. Intell.*, vol. 104, pp. 71–106, 1998.

[30] I. Miguel and Q. Shen, "Dynamic flexible constraint satisfaction," *Appl. Intell.*, vol. 13, no. 3, pp. 231–245, 2000.

Finlay S. Smith received the B.Sc. degree in mathematics and computing from the Open University, Milton Keynes, U.K., and the M.Sc. and Ph.D. degrees in artificial intelligence from Edinburgh University, Edinburgh, U.K.

He is a Lecturer in the Department of Information Technology, National University of Ireland, Galway. He has published 15 peer-refereed papers on topics within artificial intelligence and related areas in academic journals and conference proceedings. His research interests include approximate modelling, fault diagnosis, fuzzy logic, qualitative reasoning.

Qiang Shen received the B.Sc. and M.Sc. degrees in communications and electronic engineering from the National University of Defence Technology, China, and the Ph.D. degree in knowledge-based systems from Heriot-Watt University, Edinburgh, U.K

He is a Senior Lecturer in the School of Informatics, University of Edinburgh, Edinburgh, U.K., where he leads the Approximate and Qualitative Reasoning Group. He has published over 130 peer-refereed papers on topics within artificial intelligence and related areas in academic journals and conference proceedings. His research interests include fuzzy and imprecise modelling, model-based inference, pattern recognition, and knowledge refinement and reuse.

Dr. Shen is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and an Editorial Board Member of the journal, *Fuzzy Sets and Systems*.